

Claims

1. An integrated development environment (999) for developing user interface documents, comprising:  
an editor (104) for editing a user interface  
5 document (300);  
an adaptation engine (105) for generating device class specific representations (301, 302) of the user interface document (300), each device class specific representation (301, 302)  
10 referring to a respective device class (DC1, DC2);  
characterized in that  
the integrated development environment (999)  
further comprising a device class dependent  
15 complexity indicator (121) for determining complexity values of layout components (1 to 9) of the device class specific representations (301, 302) by using complexity evaluation functions (EF5-DC1, EF5-DC2, EF6-DC1, EF6-DC2), associated with the layout  
20 components (5, 6) and for aggregating the complexity values by device class according to a corresponding layout component hierarchy (321, 322) of the respective device class  
25 specific representation (301, 302).
2. The integrated development environment of claim 1, further comprising:  
a template wizard (106) being interfaced to the  
30 editor (104) for creating (502) a new user interface document (300) by loading a predefined document template from the template wizard (106) into the editor (104).

3. The integrated development environment of claim 3, where the editor (104) is interfaced to a template XML description file including information about different available document templates, the information comprising meta data about device classes supported by the templates.
4. The integrated development environment of any one of the previous claims, further comprising a tree-based outline editor (109) for generating an outline view (209) of the user interface document (300) when loaded into the editor (104), the tree-based outline editor (109) being interfaced to the editor (104) so that selection of an element (209') in the outline view 209 causes the editor (104) to highlight (504) a corresponding text portion (309) of the user interface document (300).
5. The integrated development environment of any one of the previous claims, further comprising: a code completion tool (102) for proposing possibilities for auto-insertion of text in the editor (104) dependent on document context at a specific position within the user interface document (300).
6. The integrated development environment of any one of the previous claims, further comprising: a fragment repository (123) for saving from or loading to the user interface document (300) a document fragment having a layout that is specific to a specific device class.

7. The integrated development environment of any one of the previous claims, further comprising:  
a Java filtering tool (108) for hiding Java code in the editor (104) when using an XML view for editing the user interface document (300), and for editing Java code when activating a Java code view for editing the user interface document (300), wherein the editor (104) is configured to save the user interface document (300) including Java code independent from the current editing view.
8. The integrated development environment of any one of the previous claims, further comprising:  
a device class dependent frames layouting view (124) being interfaced to the editor (104) for providing an overview of presentation structures of the user interface document (300) for various device classes.
9. The integrated development environment of any one of the previous claims, further comprising:  
a device class dependent page view (122) for using the adaptation engine (105) to execute a pre-pagination run with respect to the device class specific representations (301, 302) and for visualizing the result of the pre-pagination run for the respective device classes (DC1, DC2).

10. The integrated development environment (999) of any one of the claims from 1 to 8; wherein the device class dependent complexity indicator (121) is replaced by
- 5 a device class dependent page view (122) for using the adaptation engine (105) to execute a pre-pagination run with respect to the device class specific representations (301, 302) and for visualizing the result of the pre-
- 10 pagination run for the respective device classes (DC1, DC2).
11. The integrated development environment (999) of any one of the claims from 1 to 7; wherein the
- 15 device class dependent complexity indicator (121) is replaced by
- a device class dependent frames layouting view (124) being interfaced to the editor (104) for providing an overview of presentation
- 20 structures of the user interface document (300) for various device classes.
12. The integrated development environment claim 11, further comprising:
- 25 a device class dependent page view (122) for using the adaptation engine (105) to execute a pre-pagination run with respect to the device class specific representations (301, 302) and for visualizing the result of the pre-
- 30 pagination run for the respective device classes (DC1, DC2).

13. A computer implemented method for generating user interface documents, comprising the steps of:  
loading a user interface document (300) into an editor (104);  
5 generating device class specific representations (301, 302) of the user interface document (300) by using an adaptation engine (105), wherein each device class specific representation (301, 302) refers to a  
10 respective device class (DC1, DC2);  
characterized in that the method comprises the further steps performed by a complexity indicator (121):  
determining complexity values of layout components  
15 (1 to 9) of the device class specific representations (301, 302) by using complexity evaluation functions (EF5-DC1, EF5-DC2, EF6-DC1, EF6-DC2), associated with the layout components (5, 6); and  
20 aggregating the complexity values by device class according to a corresponding layout component hierarchy (321, 322) of the respective device class specific representation (301, 302).
- 25 14. The method of claim 13, comprising the further step:  
providing an overview of presentation structures of the user interface document (300) for various device classes.

30

15. The method of claim 13 or 14, comprising the further steps:  
executing a pre-pagination run with respect to the device class specific representations (301,  
5 302) by using the adaptation engine (105); and visualizing the result of the pre-pagination run for the respective device classes (DC1, DC2) in a device class dependent page view (122).
- 10 16. The method of claim 13 or 14, wherein the determining and aggregating steps are replaced by the steps:  
executing a pre-pagination run with respect to the device class specific representations (301,  
15 302) by using the adaptation engine (105); and visualizing the result of the pre-pagination run for the respective device classes (DC1, DC2) in a device class dependent page view (122).
- 20 17. A computer system comprising at least one computing device having data storage means and at least one processor to run an integrated development environment (999) according to any one of the claims 1 to 12.

25